WEB TECHNOLOGIES

JavaScript Object Inheritance

Vinay Joshi
Department of
Computer Science and Engineering

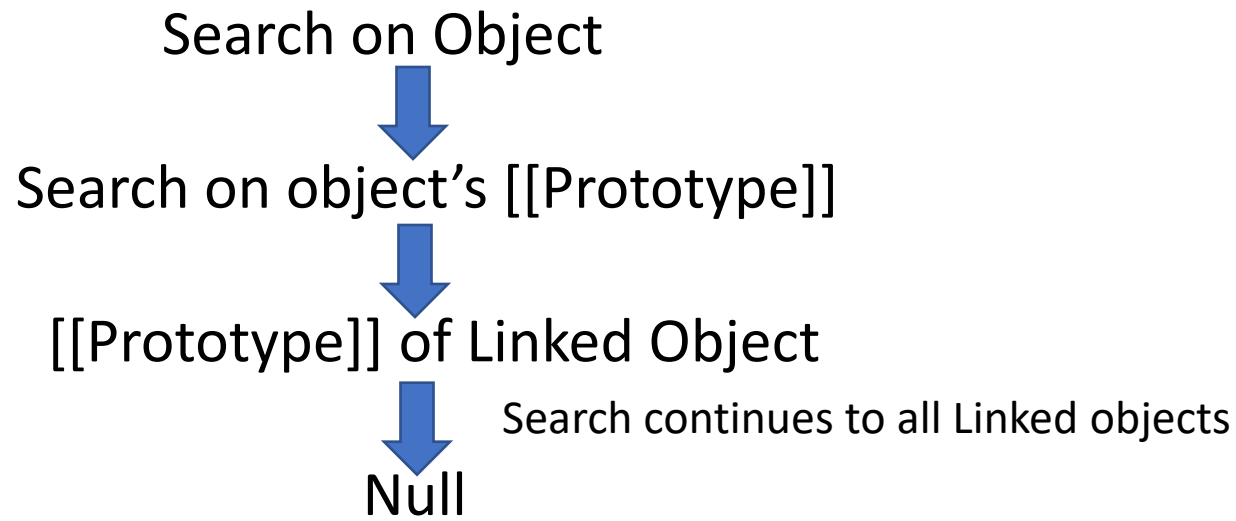**Prototype Property - Accessing Properties and Methods**

- JavaScript is a prototype based language-object properties and methods can be shared through generalized objects that have the ability to be cloned and extended.

- All objects in JavaScript descend from the parent Object constructor.

- Every object in JavaScript has an internal property called [[Prototype]]

- Built in objects also has Prototype property

- Adding new properties and methods to the prototype property is better compared to adding directly to the constructor.
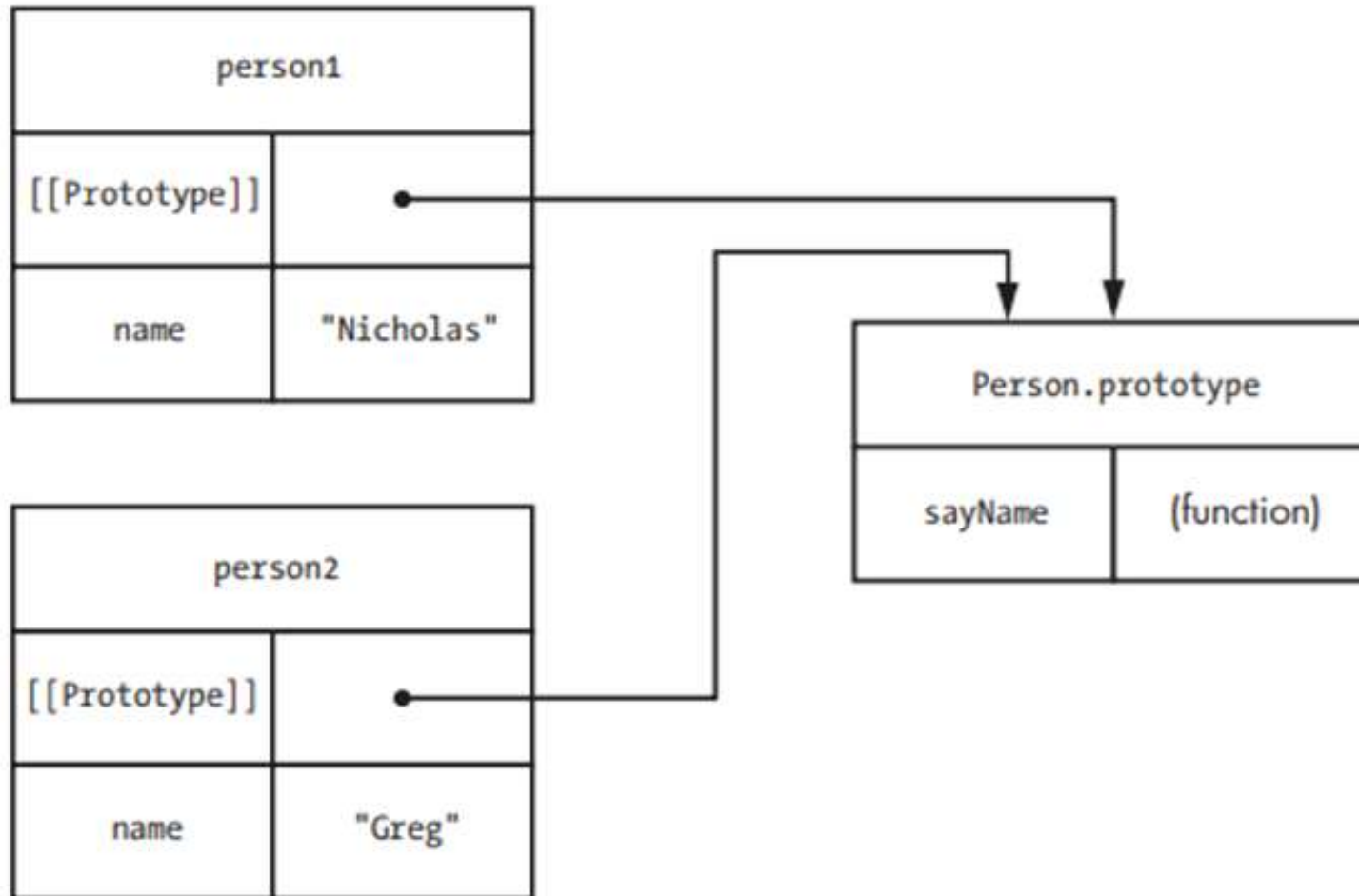
- All objects in JavaScript inherit from at least one other object.
- The object being inherited from is known as the prototype, and the inherited properties can be found in the prototype object of the constructor.
- Access properties and methods outside the objects

Search on Object

⬇

Search on object's [[Prototype]]

⬇

[[Prototype]] of Linked Object

⬇   Search continues to all Linked objects

Null

```javascript
// Let's create an object car from function
vehicle with its own properties name and year:
let Vehicle = function () {
    this.name = "Hyndai";
    this.year = 2019;
}
let car = new Vehicle();
// add properties in vehicle function's prototype
vehicle.prototype.year = 2020;
vehicle.prototype.color = "blue";
```

car.[[Prototype]] = {year,color}

car.[[Prototype]].[[Prototype]] => Object.prototype

car.[[Prototype]].[[Prototype]].[[Prototype]] =>null

- console.log(car.name)?

- console.log(car.year)? Prototype Shadowing happens here

- console.log(car.color)?

- console.log(car.model)?

- Object.create() method is used to create a new object with the

specified prototype object and properties.

- Returns a new object with the specified prototype object.

- Used for implementing inheritance.

     Object.create(prototype_object, propertiesObject)

prototypeObject: Newly created object's prototype object. It has to be an object or null.

propertiesObject: Properties of the new object. (Data Descriptors and Access Descriptors)

- Define the base object constructor

- Define the derived object constructor which in turn calls the

  base object constructor

- Assign the prototype of derived object as the base object

- Assign the prototype.constructor to derived object constructor

This will enable the derived object to access the properties and

methods of the base object.

THANK YOU

Vinay Joshi

Department of
Computer Science and Engineering

vinayj@pes.edu